


Exhibit 2

Charted Claim:
Method Claim: 1

CLAIM 1 (US8521927B2)	OnePlus Open (“Accused Product”)
1. A text entry system for an electronic device, the system comprising:	<p data-bbox="658 384 1787 411"><i>The OnePlus Open uses the OxygenOS system, which includes Gboard for text entry.</i></p> <div data-bbox="869 459 1785 970"></div> <div data-bbox="669 1034 831 1066">Performance</div> <div data-bbox="1115 1034 1749 1118">Operating System <u>OxygenOS 13.2 based on Android™ 13</u> (Emerald Dusk/Voyager Black variant) OxygenOS 14.0 based on Android™ 14 (Crimson Shadow variant)</div> <p data-bbox="658 1142 1225 1169">https://www.oneplus.com/us/oneplus-open</p>

Keyboard & input method

You can set your preferred keyboard here.

- **Current keyboard.**

- **Manage Keyboard**

- **Gboard: multilingual typing.**

- **Google voice typing:** Enable this feature to enter text by voice.

*Note: You can choose between Gboard and Google voice typing.

- **Keyboard settings:**

- **Keyboard location** (Standard/Raised)

- **Secure keyboard for passwords**

- **Mistouch prevention:** Swiping on the left or right edges of the keyboard will not return you to the previous page.

Link 1:

https://service.oneplus.com/content/dam/support/user-manuals/en/OnePlus_Open_User_Manual_EN.pdf , Page30.

Link 2:

https://service.oneplus.com/content/dam/support/user-manuals/en/OnePlus_Android_13_system_introduce.pdf , Page 27

Gboard, now available for Android

Dec 16, 2016 · 2 min read



Reena Lee
Product Manager, Gboard

Starting today, Google Keyboard on Android is getting a new name — and so much more. Gboard has all the things you love about your old Google Keyboard — speed and accuracy, [Glide Typing](#) and voice typing — plus Google Search built in. You can search and send information, GIFs, emojis and more — right from your keyboard. As an added bonus, we've added multilingual typing to help you switch languages on the fly.

Smarter and faster typing

Gboard is assisted by machine learning to make typing faster and easier, including improved [Glide Typing](#), predictions and autocorrections. And for our multilingual users, communication on a keyboard has never been easier — Gboard will now autocorrect and suggest from any of your enabled languages, so you can type in multiple languages within the same conversation without manually switching between them. Just select your languages and type.

Link: <https://blog.google/products/search/gboard-now-on-android/>

(a) an input subsystem configured to receive user input operations, and interpret the input operations as letters, characters, symbols, commands or functions, wherein said input subsystem is configured to recognize as a letter entry input operations only keystrokes performed on a plurality of keys,

The OnePlus Open uses Gboard, which includes an input subsystem designed to receive and interpret user input as letters, characters, symbols, commands, or functions. This subsystem recognizes letter entries only from keystrokes performed on a set of keys. Each keystroke corresponds to a single letter with one keystroke representing one letter location. The system supports both individual letter operations (selecting a specific letter) and letter group operations (selecting a group of possible letters). The keys include both single letter keys for individual letters and group letter keys for letter groups.

wherein each of the letter entry input operation is a single keystroke performed on a single key and each said letter entry input operation is interpreted as a single letter location entry, wherein a single letter location is entered by one and only one said letter entry input operation and wherein said letter entry input operations for each said single letter location entry is comprising both either (1) single letter operations that select any possible letter from the alphabet, or (2) letter group operations that select a group of possible letters from the alphabet, and wherein the plurality of keys comprises single letter keys for the single letter operations and group letter keys for the letter group operations;

Keyboard & input method

You can set your preferred keyboard here.

- **Current keyboard.**
- **Manage Keyboard**
 - **Gboard: multilingual typing.**
 - **Google voice typing:** Enable this feature to enter text by voice.
*Note: You can choose between Gboard and Google voice typing.
- **Keyboard settings:**
 - **Keyboard location** (Standard/Raised)
 - **Secure keyboard for passwords**
 - **Mistouch prevention:** Swiping on the left or right edges of the keyboard will not return you to the previous page.

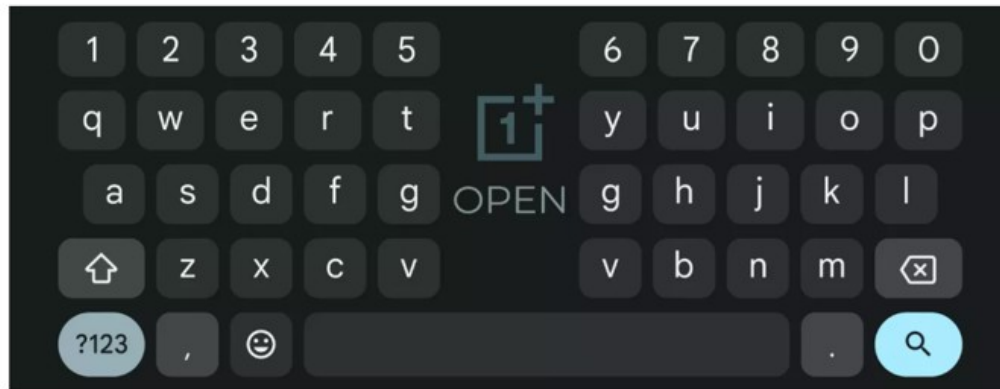
Link 1:

https://service.oneplus.com/content/dam/support/user-manuals/en/OnePlus_Open_User_Manual_EN.pdf , Page30.

Link 2:

https://service.oneplus.com/content/dam/support/user-manuals/en/OnePlus_Android_13_system_introduce.pdf , Page 27

	<div data-bbox="689 212 779 304"></div> <div data-bbox="801 212 1299 256"><h2>Gboard - the Google Keyboard</h2></div> <div data-bbox="1675 212 1709 244">×</div> <div data-bbox="801 260 952 292"><p>About this app</p></div> <div data-bbox="689 339 1653 400"><p>Gboard has everything you love about Google Keyboard—speed and reliability, Glide Typing, voice typing, Handwriting, and more</p></div> <div data-bbox="689 432 1361 464"><p>Glide Typing — Type faster by sliding your finger from letter to letter</p></div> <div data-bbox="689 496 1120 528"><p>Voice typing — Easily dictate text on the go</p></div> <div data-bbox="689 560 1193 592"><p>Handwriting* — Write in cursive and printed letters</p></div> <div data-bbox="689 624 1081 655"><p>Emoji Search* — Find that emoji, faster</p></div> <div data-bbox="689 687 1243 719"><p>GIFs* — Search and share GIFs for the perfect reaction.</p></div> <div data-bbox="689 751 1675 812"><p>Multilingual typing — No more switching between languages manually. Gboard will autocorrect and suggest from any of your enabled languages.</p></div> <div data-bbox="656 818 1854 850"><p>Link: https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin</p></div>
--	--

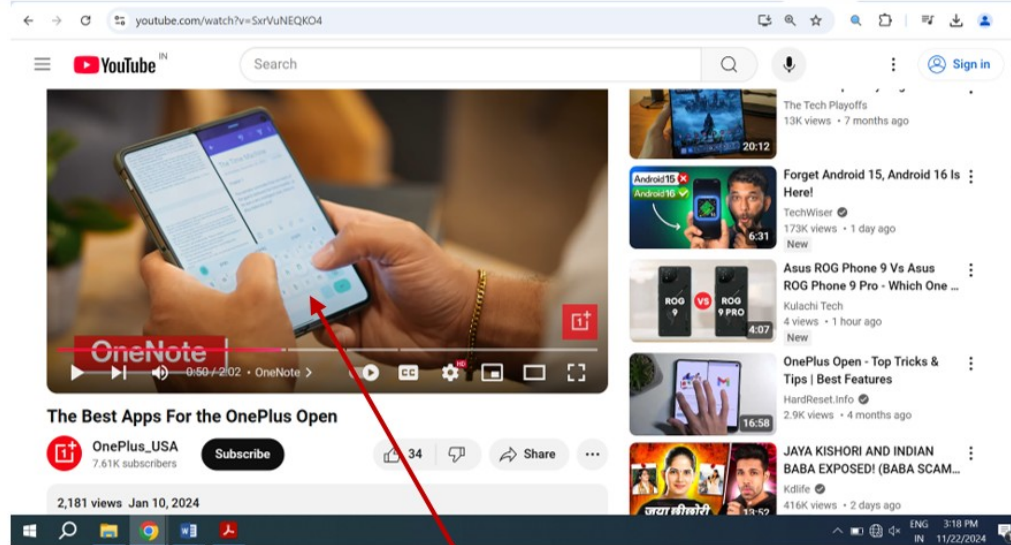


With the large 7.82" Flexi-fluid AMOLED display, multi-tasking has become so much easier! Be it working on an article/moment/poll on the Community, assignments, editing pictures/documents/presentations, streaming the Cricket World Cup, planning your next trip, *asking ChatGPT for help*, or a multitude of tasks!

One of the most commonly used features while multitasking is the KEYBOARD! To assist the users, the Open offers three orientations, as shown below.

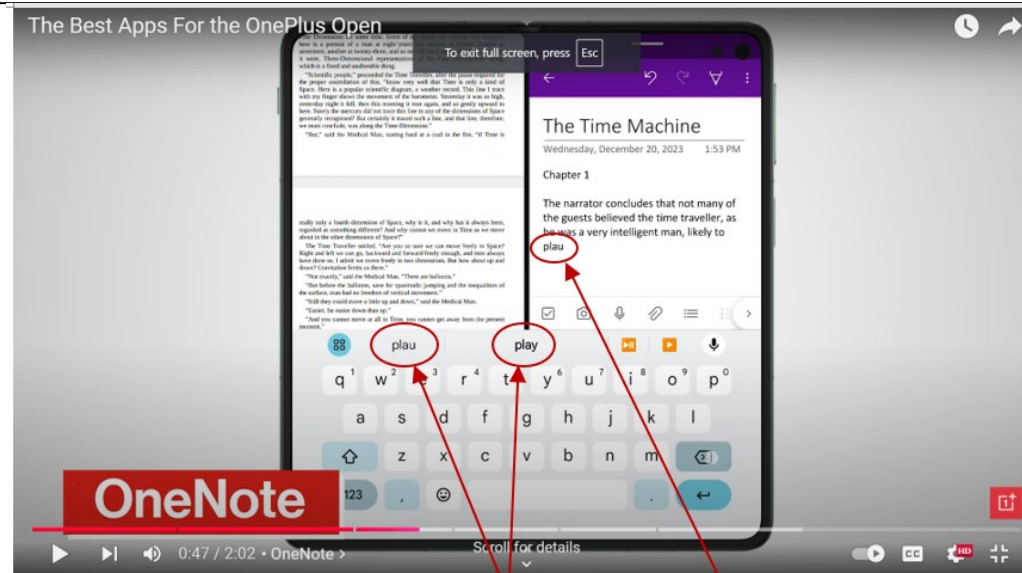
I'd like to invite y'all to share your preferred orientation - Split, Standard, or Floating Keyboard ☐

Link: <https://community.oneplus.com/thread/1446357664756924423>



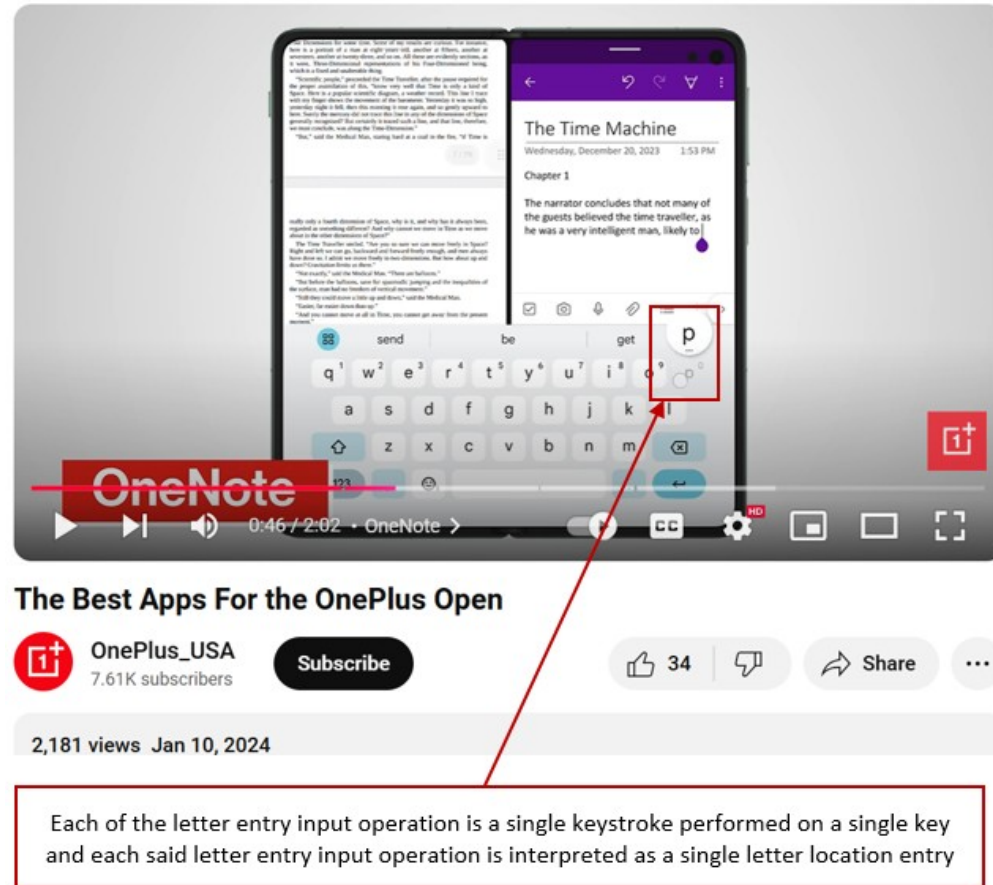
an input subsystem configured to receive user input operations, and interpret the input operations as letters, characters, symbols, commands or functions.

Link: <https://www.youtube.com/watch?v=SxrvuNEQKO4>

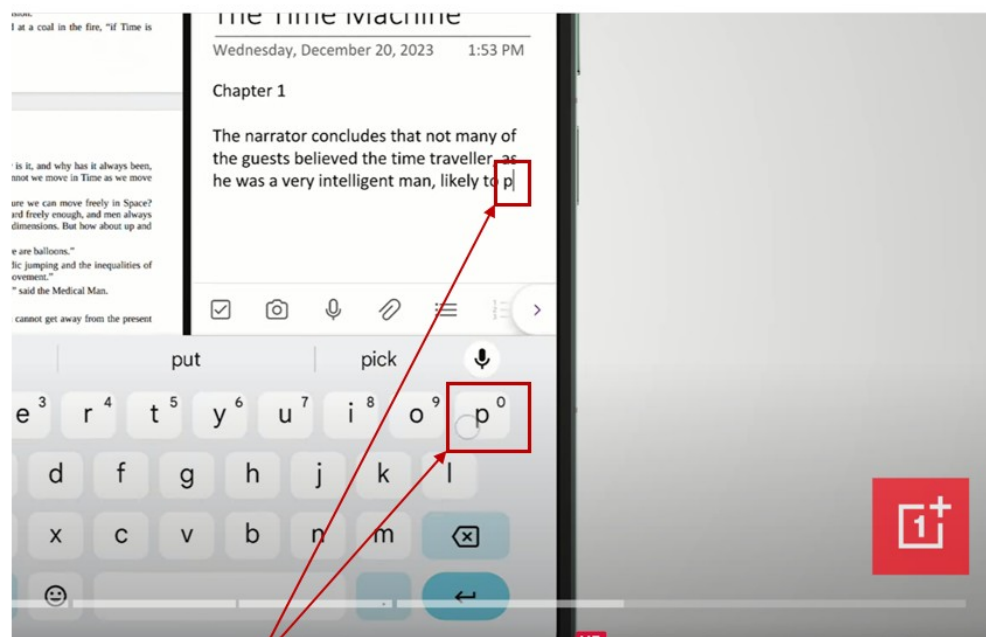


Input subsystem is configured to recognize as a letter entry input operations only keystrokes performed on a plurality of keys

Link: <https://www.youtube.com/watch?v=SxrVuNEQKO4>

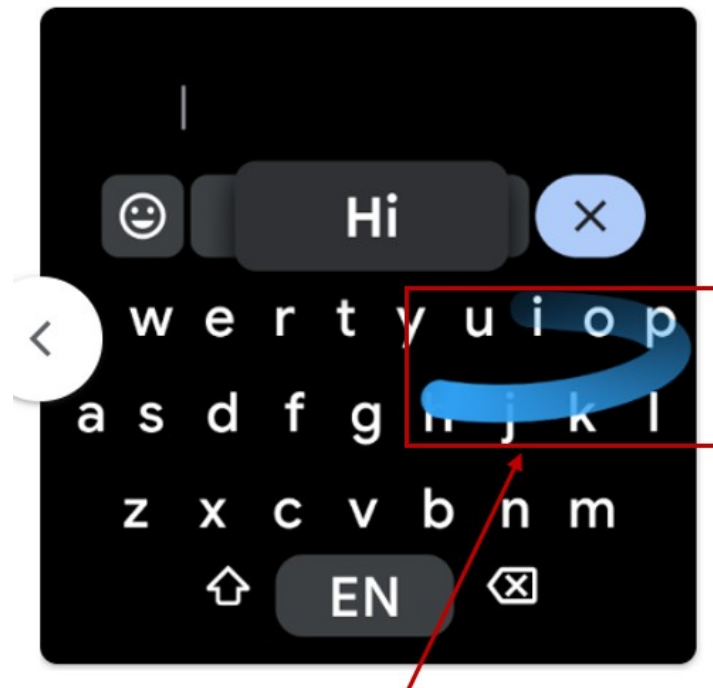


Link: <https://www.youtube.com/watch?v=SxrvuNEQKO4>



A single letter location is entered by one and only one said letter entry input operation

Link: <https://www.youtube.com/watch?v=SxrVuNEQKO4>

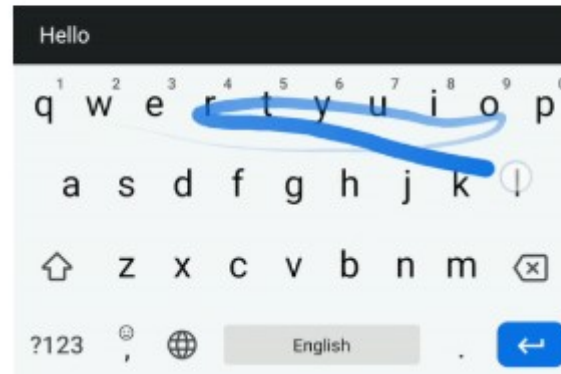


The plurality of keys comprises group letter keys for the letter group operations.

Link:

https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin&hl=en_US


Glide type words and phrases



Help improve [Google Keyboard's](#) ability to better recognize words typed via glide typing (sliding finger from letter to letter), thereby helping Gboard users type with more accuracy and speed.

When you glide type the prompted text, your typing sequence (gesture trail), word suggestions and the word you submit are recorded. Google may use this data to improve and develop Google products, services and machine learning technology, such as the ability to better recognize words and phrases when users glide type on Gboard.

Follow these guidelines when doing this activity.

- Set [Gboard](#) as the primary keyboard on your phone.
- Slide your finger from letter to letter to type the prompted text (Glide typing). Type the text exactly as it is written, including spelling mistakes, punctuation marks and spaces between words. **Do not tap type.**
- If a word is incorrectly recognized, delete and re-type.
- If you find a word too difficult to glide type, tap **Skip**.
- Note that the task is case-insensitive.
- If you believe a set of words is inappropriate or offensive for any reason, tap More  > Flag.

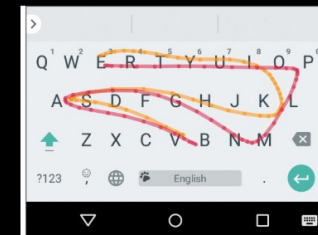
Link: <https://support.google.com/crowdsource/answer/10279623?hl=en>

[Home](#) > [Blog](#) >

The Machine Intelligence Behind Gboard

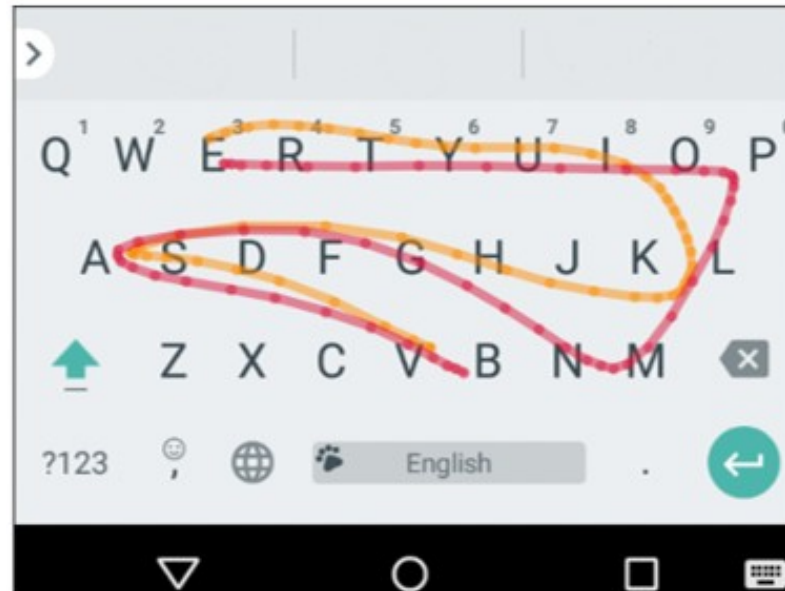
May 24, 2017 ·

Posted by Françoise Beaufays, Principal Scientist, Speech and Keyboard Team and Michael Riley, Principal Scientist, Speech and Languages Algorithms Team



Neural Spatial Models

Mobile keyboard input is subject to errors that are generally attributed to “fat finger typing” (or tracing spatially similar words in glide typing, as illustrated below) along with cognitive and motor errors (manifesting in misspellings, character insertions, deletions or swaps, etc). An intelligent keyboard needs to be able to account for these errors and predict the intended words rapidly and accurately. As such, we built a spatial model for Gboard that addresses these errors at the character level, mapping the touch points on the screen to actual keys.

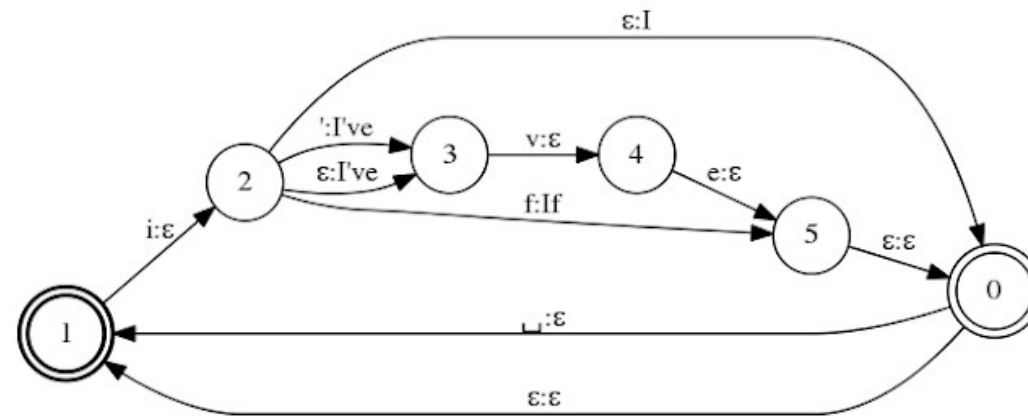


Average glide trails for two spatially-similar words: “Vampire” and “Value”.

Finite-State Transducers

While the NSM uses spatial information to help determine what was tapped or swiped, there are additional constraints — *lexical* and *grammatical* — that can be brought to bear. A lexicon tells us what words occur in a language and a probabilistic grammar tells us what words are likely to follow other words. To encode this information we use finite-state transducers. FSTs have long been a key component of Google’s speech recognition and synthesis systems. They provide a principled way to represent various probabilistic models (lexicons, grammars, normalizers, etc) used in natural language processing together with the mathematical framework needed to manipulate, optimize, combine and search the models*.

In Gboard, a key-to-word transducer compactly represents the keyboard lexicon as shown in the figure below. It encodes the mapping from key sequences to words, allowing for alternative key sequences and optional spaces.



This transducer encodes “I”, “I’ve”, “If” along paths from the start state (the bold circle 1) to final states (the double circle states 0 and 1). Each arc is labeled with an input key (before the “:”) and a corresponding output word (after the “:”) where ϵ encodes the empty symbol. The apostrophe in “I’ve” can be omitted. The user may skip the space bar sometimes. To account for that, the space key transition between words in the transducer is optional. The ϵ and space back arcs allow accepting more than one word.

A probabilistic n-gram transducer is used to represent the language model for the keyboard. A state in the model represents an (up to) n-1 word context and an arc leaving that state is labeled with a successor word together with its probability of following that context (estimated from textual data). These, together with the spatial model that gives the likelihoods of sequences of key touches (discrete tap entries or continuous gestures in glide typing), are combined and explored with a [beam search](#).

Link: <https://research.google/blog/the-machine-intelligence-behind-gboard/>

Neural Search Space in Gboard Decoder

Yanxiang Zhang*, Yuanbo Zhang*, Haicheng Sun, Yun Wang, Billy Dou,
Gary Sivek, Shumin Zhai
Google Inc
zhangyx,zyb,haicsun,wyun,billydou,gsivek,zhai@google.com

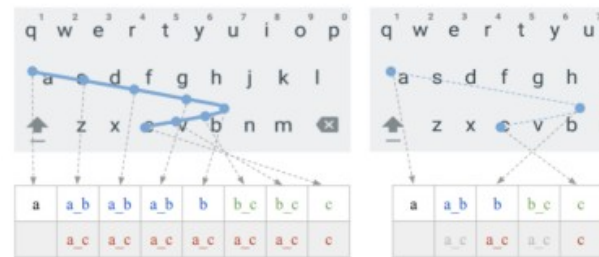
1 Introduction

Gboard is a statistical-decoding-based keyboard on mobile devices developed by Google. Statistical decoding is far more necessary than one might think due to the error-prone process of “fat finger” touch input on small screens. According to [Azenkot and Zhai \(2012\)](#), the per-letter error rate is around 8%-9% without decoding. With decoding, typos such as substitutions (due to the proximity of two keys or cognitive misspellings), omissions, insertions, and transpositions could be automatically corrected by the key-correction and (word) auto-correction functions in the Gboard decoder, leading to an error-tolerant user experience. Powered by language models (LM), the Gboard decoder also

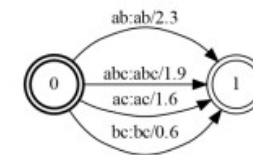
provides rich functionalities such as word completion, post correction, next word prediction, smart compose (in-line predictions) to further save users' physical input effort.

The decoding process involves two phases: building search space (decoder graph), and performing beam search within the space based on user touch inputs. Gboard decoder utilizes context, a lexicon and language transducers - the familiar $C \circ L \circ G$ composition (Ouyang et al., 2017; Hellsten et al., 2017) - to construct the search space. C is a bi-key key to key transducer while L is a key to word transducer, C and L are statically composed together offline since the size is small. Fig. 1-A illustrates how gesture typing and tap typing inputs are converted into bi-keys and Fig. 1-B illustrates a composed $C \circ L$ targeting four words. Before this work, G is a N-gram language FST containing 64k words for n-grams and 170k words for uni-grams. Composition between $(C \circ L)$ and G are dynamically conducted due to the large size of G . Fig. 1-C

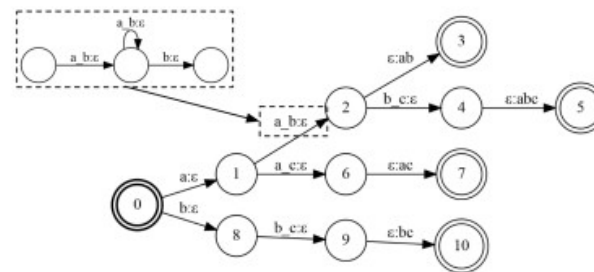
shows a simple G containing only four words, and Fig. 1-D illustrates a composed $(C \circ L) \circ G$, which is similar to $(C \circ L)$ but with weights achieved by using the look-ahead composition filters proposed by Allauzen et al. (2009, 2011).



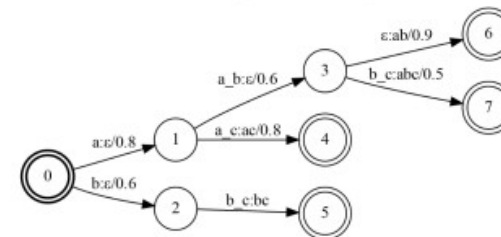
(A). Gesture / Tapping to bi-keys



(C). G , a simple LM with unigrams



(B). $C \circ L$, which is statically composed



(D). $(C \circ L) \circ G$, composed at runtime, label push / weight push are applied.

Figure 1: Build search space by composing $(C \circ L) \circ G$

Link: <https://aclanthology.org/2024.emnlp-industry.93.pdf>

MOBILE KEYBOARD INPUT DECODING WITH FINITE-STATE TRANSDUCERS

Tom Ouyang, David Rybach, Françoise Beaufays, Michael Riley

Google

{ouyang, rybach, fsb, riley}@google.com

3. KEYBOARD TRANSDUCERS

A tapped input consists of a time series of touch points, \mathbf{x} , that encodes the coordinates of the user's key presses. For gesture input, the input trajectory is sampled, e.g. every 100 milliseconds, to provide a similar time series. The task of the decoder is to find the word sequence \mathbf{w} that best matches the input sequence \mathbf{x} .

3.1. Key Context Dependency and Spatial Model

The equivalent of speech phonemes in the keyboard world is the set of keys offered in the layout. Accordingly, a spatial model is used to provide a probability distribution over these units. Note that the spatial model does not resolve fully the written language: For example, the letters “é” and “è” in a French keyboard are typically obtained by long-pressing “e” and choosing from a small pop-up menu. All three letters have the same spatial score. This confusability is mitigated by imposing context dependency constraints and enforcing a strong language model.

Just like acoustic context dependency is encoded in speech with a C transducer, we implemented spatial context dependency in keyboard, and chose to do so with a bi-key model. Accordingly, the arcs of the transducer represent $a.b : b$ transitions, where $a.b$ means the key b with a as left context (previous key). This places keyboard somewhere in between our server-based speech recognizer which relies on triphone models and our embedded recognizer which uses monophones [8].

The spatial model for tap input is typically a Gaussian distribution centered on each key center. Gesture inputs instead are often modeled with the so-called “minimum-jerk model” that imposes smoothness maximization constraints on the input trajectory [9]. Alternatively, a recurrent neural network model can be used [10].

3.2. Lexicon

The lexicon transducer L for the keyboard decoder is a simple key to word mapping, like a speech grapheme lexicon. Some keys like the apostrophe may be made optional, allowing users to type “Ive” for “I’ve”. Similarly, repeated keys can be defined optional, e.g. the second “o” in “Google”. The closure, which allows transducing sequences of words, is implemented with a space symbol for finger lift-up or taps on the space bar between words. If this space symbol is optional, the decoder can correct missing space taps between words or decode multi-word gestures. An example of a lexicon FST is shown in Figure 1.

3.3. Language Model

Similar to the language models in embedded speech recognition systems, keyboard language models are typically low order n-grams over a limited vocabulary, e.g. 64K words [8].

Because of features like suggestions, completions and predictions where the language model is more prominent than the spatial model, the language model should be carefully crafted. For example, the user who gestures “Google” may expect the keyboard to suggest “goggle” as an alternative, but not “gogle” or “gooogle”, which chances are would be present in the training corpus as training corpora are often noisy. For this reason, keyboard language models are typically trained to a fixed vocabulary that has been hand-curated to eliminate misspellings, erroneous capitalizations, and other undesired artifacts.

With this, the decoder graph for keyboard is constructed using the composition of context, lexicon, and language transducers - the familiar $C \circ L \circ G$. For memory efficiency, we use the on-the-fly composition of $(C \circ L) \circ G$ using look-ahead composition filters [11].

Figure 2A shows a gesture input with its possible alignments to a simplistic, two-word decoder graph. Figure 2B shows the input sequence and its alignment for a similar tap sequence. The same decoder graph (Figure 2C) is shared for both types of input.

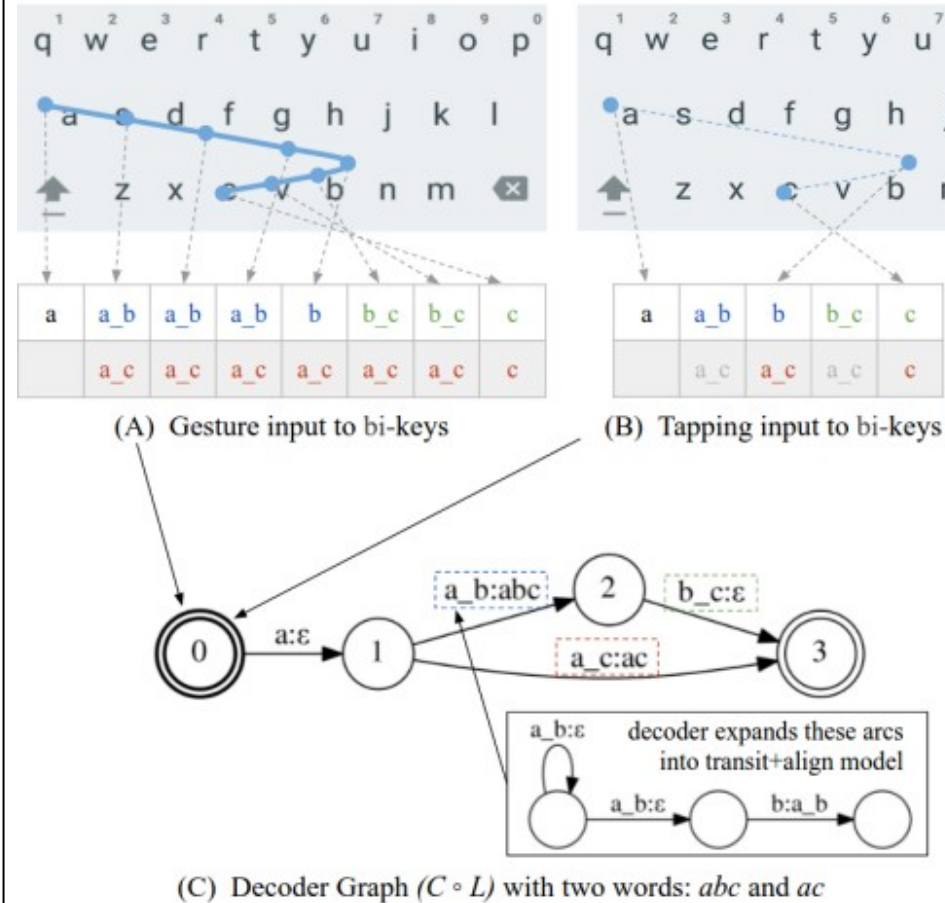


Fig. 2: A simple example of gesture (A) and tap input (B), with their state alignment to a 2-word toy FST (C) allowing only the words “abc” and “ac”.

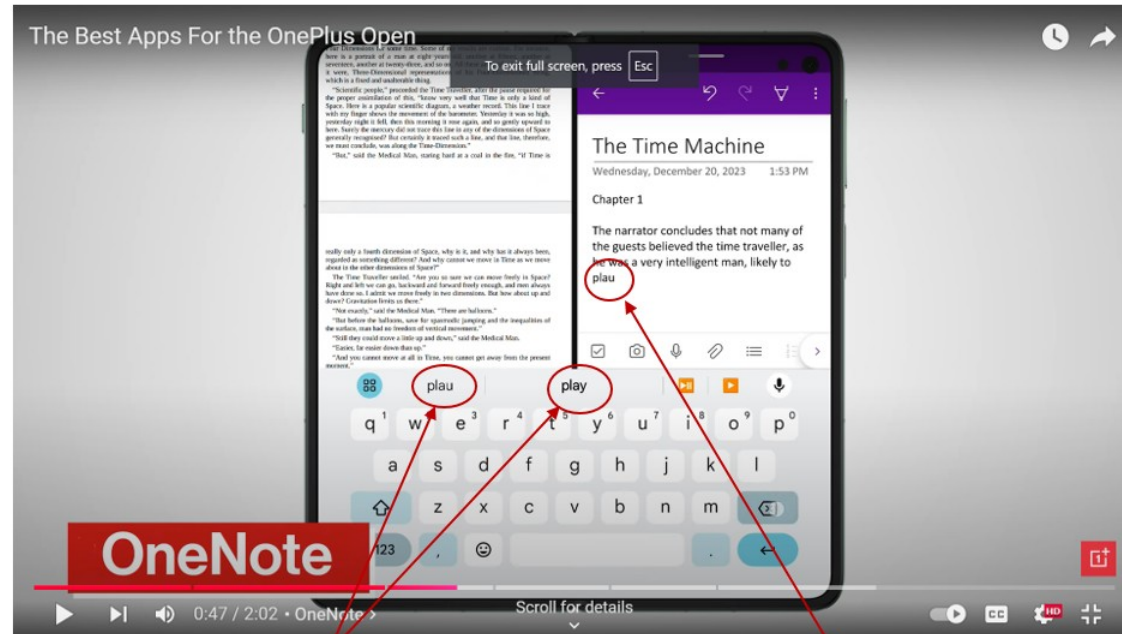
[Link: <https://arxiv.org/pdf/1704.03987>]

(b) a text prediction subsystem which receives a sequence

The OnePlus Open uses Gboard, which features a text prediction subsystem. This subsystem processes both single letter and letter group operations, producing a list of possible words based on

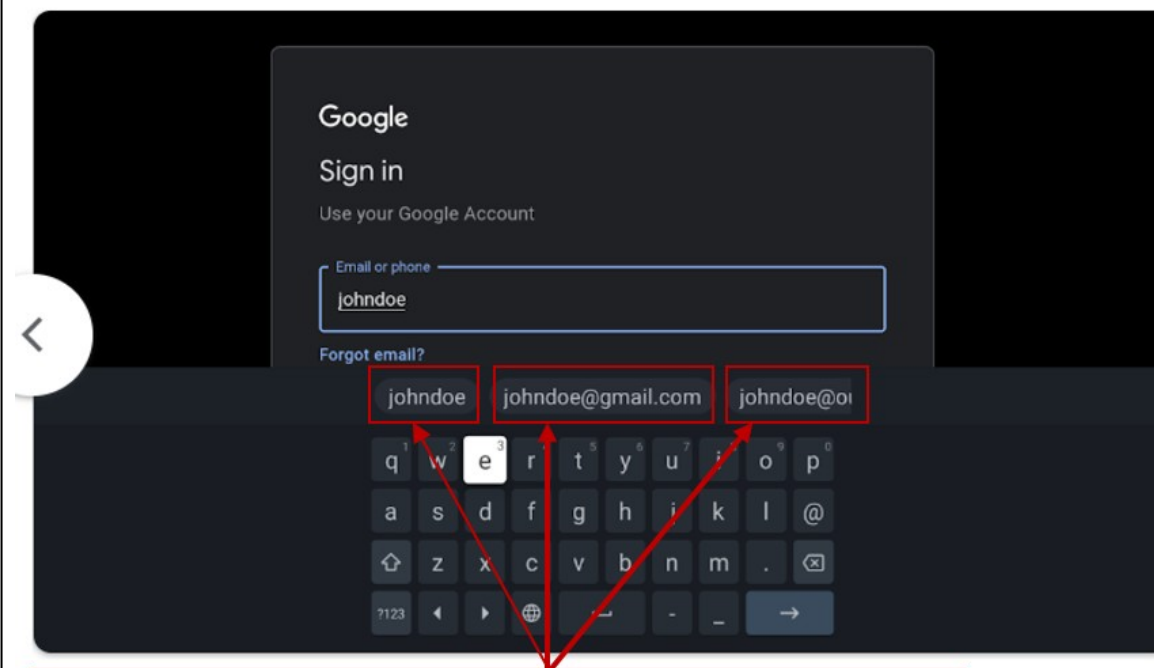
including both of said single letter operations and of said letter groups operations, and produces a list of possible words based on an a priori database of words, wherein said text prediction allows any operation of said sequence to be a single letter operation or a letter group operation, wherein said text prediction subsystem operates by searching said database for words which have letters that match letters entered by said single letter operations and resolve the ambiguity of letter groups entered by said letter group operations; and

a pre-existing word database. It matches letters entered through single keystrokes and resolves ambiguities in letter groups to predict words accurately.



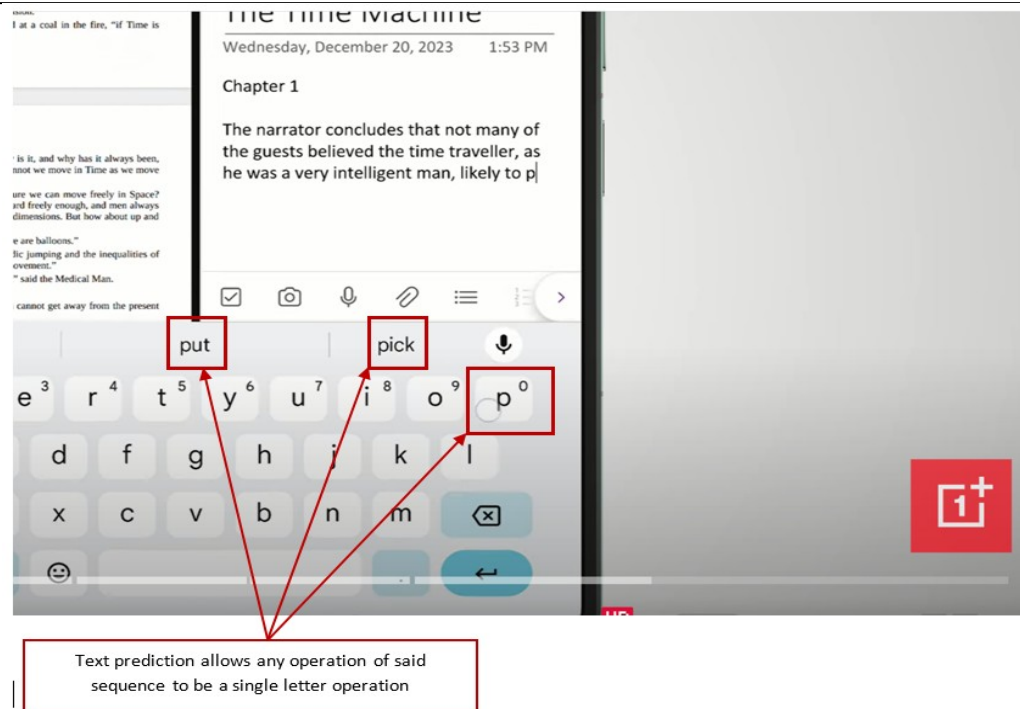
A text prediction subsystem which receives a sequence including both of said single letter operations and of said letter groups operations, and produces a list of possible words based on an a priori database of words.

Link: <https://www.youtube.com/watch?v=SxrvuNEQK04>



Produces a list of possible words based on an a priori database of words

Link: https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin&hl=en_GB



Link: <https://www.youtube.com/watch?v=SxrvuNEQK04>

[Submitted on 8 Nov 2018 (v1), last revised 28 Feb 2019 (this version, v2)]

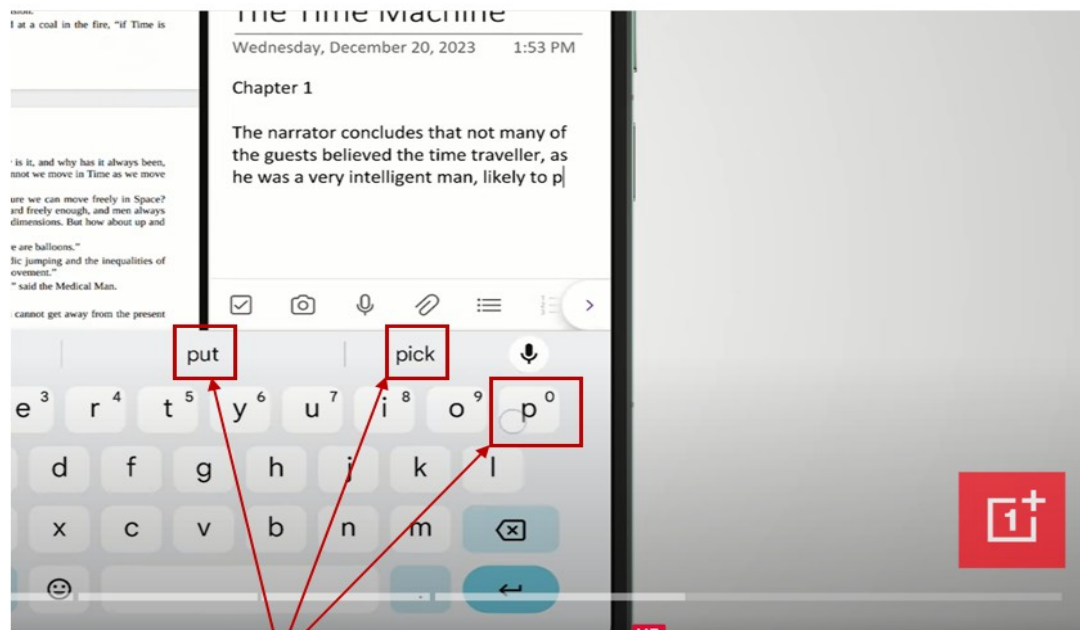
Federated Learning for Mobile Keyboard Prediction

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, Daniel Ramage

We train a recurrent neural network language model using a distributed, on-device learning framework called federated learning for the purpose of next-word prediction in a virtual keyboard for smartphones. Server-based training using stochastic gradient descent is compared with training on client devices using the Federated Averaging algorithm. The federated algorithm, which enables training on a higher-quality dataset for this use case, is shown to achieve better prediction recall. This work demonstrates the feasibility and benefit of training language models on client devices without exporting sensitive user data to servers. The federated learning environment gives users greater control over the use of their data and simplifies the task of incorporating privacy by default with distributed training and aggregation across a population of client devices.

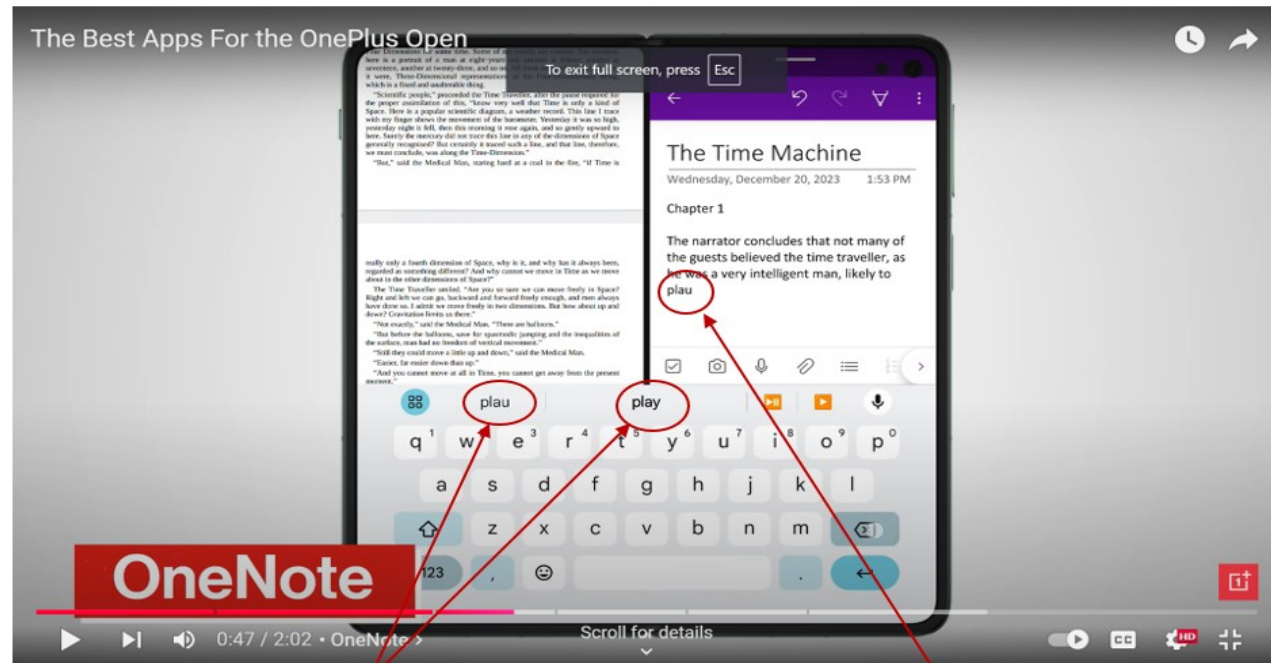
Link: <https://arxiv.org/abs/1811.03604>

	<p>Gboard's LMs are designed to work with a predefined list of frequently-used words, referred to as vocabulary. The performance of LMs depends on the quality of this vocabulary, which could change over time. Words that are not part of the vocabulary are referred to as out-of-vocabulary (OOV) words. OOV words can occur in Gboard for several reasons. For example, the vocabulary for some languages is still under development in Gboard, so the fraction of OOV words can be higher. For languages where Gboard has a relatively complete vocabulary, such as English in the U.S., OOV words often appear due to newly emerged trending words, such as "COVID-19" and "Wordle", atypical capitalization, such as "tuesday", and unusual spelling due to users' preferences, such as "coool", or even typos. OOV discovery is a challenging task due to the sensitive nature of information that users type on their keyboard.</p> <p>Today, we are excited to share a number of approaches that improve the performance of LMs by enabling the discovery of new frequently-used words, while maintaining strong data minimization and DP guarantees. These research efforts include collaboration with linguists to surface novel OOV words, employment of privacy-preserving federated analytics and other DP algorithms, and use of trusted execution environments (TEEs).</p> <p>Link: https://research.google/blog/improving-gboard-language-models-via-private-federated-analytics/</p>
(c) a word processing subsystem, which receives said list of possible words, displays said list to the user, receives a user selection of a desired word and provides the desired word as a text entry for further processing.	<p><i>The OnePlus Open uses Gboard, which includes a word processing subsystem. This subsystem displays a list of predicted words to the user, allows them to select a desired word, and then provides the selected word for further processing.</i></p>



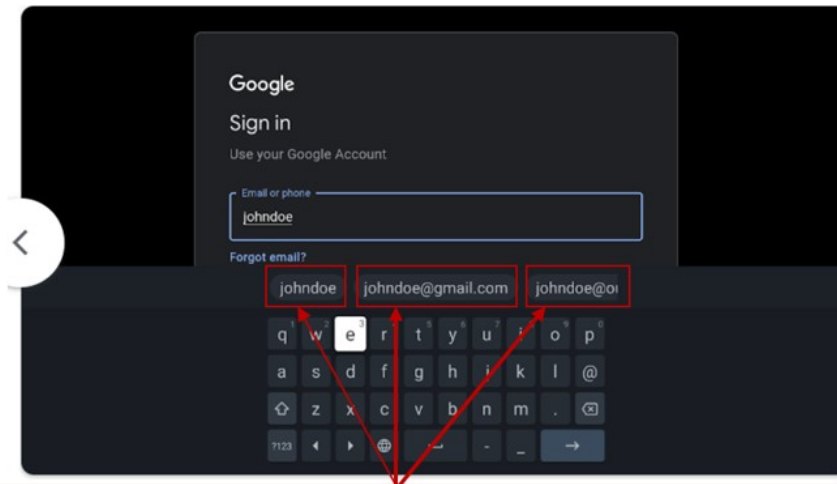
A word processing subsystem, which receives said list of possible words, displays said list to the user, receives a user selection of a desired word and provides the desired word as a text entry for further processing.

Link: <https://www.youtube.com/watch?v=SxrvuNEQKO4>



A word processing subsystem, which receives said list of possible words, displays said list to the user, receives a user selection of a desired word and provides the desired word as a text entry for further processing.

Link: <https://www.youtube.com/watch?v=SxrvuNEQK04>



A word processing subsystem, which receives said list of possible words, displays said list to the user, receives a user selection of a desired word and provides the desired word as a text entry for further processing.

Link: https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin&hl=en_GB

Get started with Gboard

Important:

- Talk-to-text doesn't work with all languages.
 - Some of these steps work only on Android 7.0 and up. [Learn how to check your Android version.](#)
1. On your Android phone or tablet, [install Gboard](#).
 2. Open any app that you can type with, like Gmail or Keep.
 3. Tap where you can enter text.
 4. Type a word. At the top of the keyboard, you'll see suggestions:
 - If you see the word you want, tap it.
 - If you don't like a suggested word, touch and hold it, and then drag the word to Trash.

Link: <https://support.google.com/gboard/answer/7068415?hl=en&co=GENIE.Platform%3DAndroid#:~:text=Tap%20where%20you%20can%20enter,word%20you%20want%2C%20tap>

[%20it](#)

Neural Search Space in Gboard Decoder

Yanxiang Zhang*, Yuanbo Zhang*, Haicheng Sun, Yun Wang, Billy Dou,
Gary Sivek, Shumin Zhai
Google Inc
zhangyx,zyb,haicsun,wyun,billydou,gsivek,zhai@google.com

Abstract

Gboard Decoder produces suggestions by looking for paths that best match input touch points on the context aware search space, which is backed by the language Finite State Transducers (FST). The language FST is currently an N-gram language model (LM). However, N-gram LMs, limited in context length, are known to have sparsity problem under device model size constraint. In this paper, we propose **Neural Search Space** which substitutes the N-gram LM with a Neural Network LM (NN-LM) and dynamically constructs the search space during decoding. Specifically, we integrate the long range context awareness of NN-LM into the search space by converting its outputs given context, into the language FST at runtime. This involves language FST structure redesign, pruning strategy tuning, and data structure optimizations. Online experiments demonstrate improved quality results, reducing Words Modified Ratio by [0.26%, 1.19%] on various locales with acceptable latency increases. This work opens new avenues for further improving keyboard decoding quality by enhancing neural LM more directly.

Link: <https://aclanthology.org/2024.emnlp-industry.93.pdf>.

--	--